

## 3

### MoLIC

Este capítulo apresenta a linguagem MoLIC considerando a sua última revisão [Araujo, 2008]. Contudo, o foco deste capítulo se dá no diagrama de interação e sua característica de apreender a visão geral da interação de um sistema, procurando dar mais ênfase no conjunto básico de elementos, mais do que em extensões propostas por trabalhos anteriores em termos de **notação**, como apresentado na seção 3.1.

A MoLIC – *Modeling Language for Interaction as Conversation* – foi inicialmente proposta por Paula [Barbosa e Paula, 2003] e continuada por Silva [Silva, 2005] e Araújo [Araujo, 2008] através de propostas de extensões. Ela foi criada para servir como ferramenta epistêmica, dando suporte à reflexão do designer acerca da solução interativa sendo concebida por ele, partindo do ponto de vista de que ele tem o usuário como interlocutor em uma conversa.

Como ferramenta epistêmica, ela foi criada principalmente para ajudar em decisões de design, mais no sentido de aumentar o conhecimento do designer sobre o problema que este tem em mãos, do que para derivar soluções diretas para o problema ou ainda criar especificações. O objetivo primordial da MoLIC é apoiar a reflexão sobre o artefato que está sendo projetado e o que é comunicado por ele.

A linguagem trata a interação como uma conversa entre o usuário e o sistema, que é na realidade referenciado como o preposto do designer na conversa. Para a Engenharia Semiótica, a interface é uma mensagem elaborada pelo designer com a qual o usuário tem de lidar para atingir seus objetivos [de Souza, 2005]. Se a considerarmos integrando um processo de desenvolvimento, podemos vê-la como uma ponte entra a etapa de análise de requisitos dos usuários e o projeto e construção da interface de sistemas interativos [Araujo, 2008].

A última versão da linguagem é composta de três artefatos inter-relacionados: o diagrama de interação, a lista de metas, e o esquema conceitual de signos. O foco deste trabalho se dá no diagrama de interação, mais do que

nos outros dois artefatos – a ferramenta MoLIC Designer é primariamente uma ferramenta de projeto e diagramação da interação.

### 3.1

#### Diagrama de Interação

Na etapa de modelagem do diagrama, o designer especifica a interação usuário-sistema de fato, definindo de que modo as metas dos diferentes papéis de usuários podem ser alcançadas durante a interação. A conversa entre designer e usuário deve conter todos os assuntos e diálogos possíveis entre eles, mas sem apresentar detalhes de interface ou de plataforma tecnológica. Ao construir o diagrama de interação, o designer deve modelar todos os possíveis caminhos de interação que foi capaz de prever, inclusive as eventuais rupturas de comunicação entre os interlocutores.

Podemos ilustrar a notação diagramática da MoLIC na Figura 3.1, que mostra um diagrama simples para um sistema de busca hipotético.

PUC-Rio - Certificação Digital Nº 0812613/CA

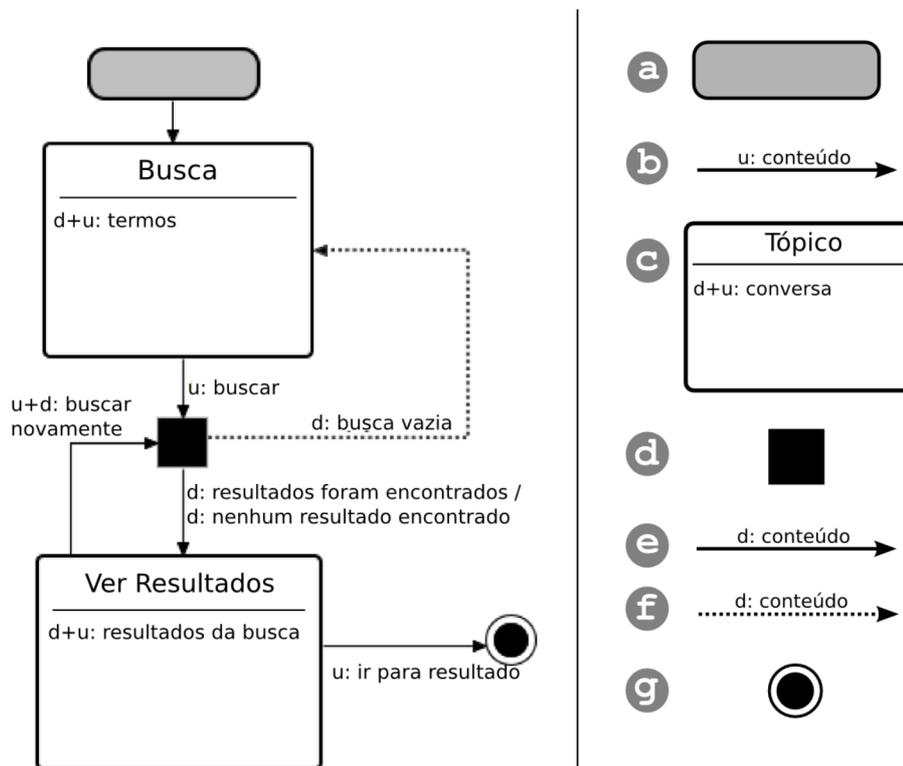


Figura 3.1: A notação da linguagem MoLIC

Os elementos básicos da linguagem podem ser assim descritos:

- a. **Acesso Ubíquo:** a oportunidade para o usuário mudar o tópico da conversa, partindo de qualquer outra cena da aplicação, geralmente para

atingir um objetivo diferente do atual. Para o sistema de busca em questão, pode significar o acesso do usuário a uma parte específica para busca;

- b. **Fala do Usuário:** ilustra a intenção do usuário de prosseguir para um determinado tópico na conversa, formatado como *u:conteúdo*. No exemplo, pode significar a expressão do usuário para a ação “buscar” (*u:buscar*);
- c. **Cena:** um momento na interação em que o usuário decide sobre como a conversa deve proceder. O primeiro compartimento contém o tópico da conversa, o segundo contém os diálogos que trazem o foco da conversa para o determinado tópico. O tópico da cena pode ser visto como o designer falando ao usuário: “Neste momento, você pode *\*tópico\**”.
- d. **Processo:** uma caixa preta representando o processamento interno do sistema que fica escondido dos usuários, alheios ao que acontece no nível imediatamente posterior ao da interface. Um aspecto fundamental dessa notação é que o usuário só irá saber o que acontece **dentro** do sistema se o preposto do designer o informar (também através de falas).
- e. **Fala do designer:** a vez do designer falar (formatado como *d:conteúdo*), geralmente em resposta a uma requisição do usuário.;
- f. **Ruptura:** indica uma fala do designer para uma situação de recuperação de ruptura, através de uma linha tracejada ;
- g. **Fechamento:** representa o fim de uma conversa (p.ex. saída do sistema).

As falas podem conter em sua sintaxe, ainda, pré-condições (representadas por exemplo como *precond: usuário deve estar logado*), pressuposições (representadas como p. ex. *presup: usuário novato*) e efeito perlocutório (ou pós-condição, representadas como p. ex. *perloc: usuário logado*)

### 3.1.1

#### Rupturas

Norman, em seu livro *The Design of Everyday Things* incentiva o tratamento de erros (aqui também referidos como rupturas na interação) como parte do processo de design (chamou de *Design for Error* pp.200), argumentando que erros são parte do diálogo natural entre usuário e sistema. Assim,

o designer deve **apoiar**, e não **lutar contra** as respostas dos usuários, permitindo que estes possam se recuperar de situações de erro, sabendo claramente o que aconteceu e como reverter a situação [Norman, 1988].

As falas de recuperação de rupturas são o diferencial mais significativo da MoLIC com relação às outras maneiras de modelar interação, uma vez que as linguagens para modelagem de tarefas geralmente especificam a interação como ela deve ocorrer no melhor caso, onde o usuário acerta todos os passos e se comporta exatamente como o previsto pelo modelo.

Na vida real, porém, usuários cometem erros como parte dos passos para atingir seus objetivos. A linguagem MoLIC incentiva a consideração, por parte do designer, desses caminhos alternativos através da modelagem ativa de rupturas. Assim, o modelo pode contemplar não apenas os caminhos preferidos de interação, mas também caminhos para apoiar o usuário em situações mal-sucedidas.

A MoLIC considera cinco tipos de mecanismos de tratamento ou prevenção de rupturas comunicativas[Barbosa e Paula, 2003]:

- a. **Prevenção Passiva (PP - *Passive Prevention*)**: no caso de erros que podem ser evitados por documentação ou por instruções online, por exemplo comunicando ao usuário previamente “Todos os campos marcados com \* são obrigatórios”;
- b. **Prevenção Ativa (AP - *Active Prevention*)**: no caso de erros que são ativamente evitados pelo sistema, impedindo que o usuário produza uma fala inválida que o levaria a uma situação de erro. Uma implicação pode ser gerar uma especificação na interface onde o botão OK é desabilitado quando um campo do formulário não está digitado corretamente, por exemplo;
- c. **Prevenção Apoiada (SP - *Supported Prevention*)**: para situações em que o sistema detecta como passíveis de erro em potencial, mas em que a decisão está a cargo do usuário, como por exemplo “O Arquivo XXX já existe, sobrescrever?”;
- d. **Recuperação Apoiada (SR - *Supported Recovery*)**: quando o designer deve contar ao usuário de que forma ele pode se recuperar de um erro e oferece a oportunidade para que ele o faça, p.ex. contando ao usuário o que aconteceu de errado e apresentando o momento da conversa em que aconteceu a ruptura, para que o usuário possa proceder com a conversa.

- e. **Captura de Erro (EC - *Error Capture*)**: erros que são identificados pelo sistema mas não são recuperados **através** do sistema. Quando o preposto é capaz de identificar erros do sistema, deve notificá-los ao usuário e, se possível, sugerir formas de se recuperar do erro. Por exemplo “Não foi possível conectar com o servidor, deve haver algum problema com a sua conexão com a Internet”;

## 3.2

### Lista de Metas

A lista de metas relaciona as metas identificadas na etapa de análise com os diferentes papéis de usuário da aplicação, especificando o quê os usuários podem realizar com o sistema. A proposta inicial da MoLIC envolvia representar uma meta, no diagrama de **interação**, basicamente por um agrupamento que circunscreve (graficamente) os elementos conectados que tenham em comum o atingimento de uma determinada meta (Figura 3.2).

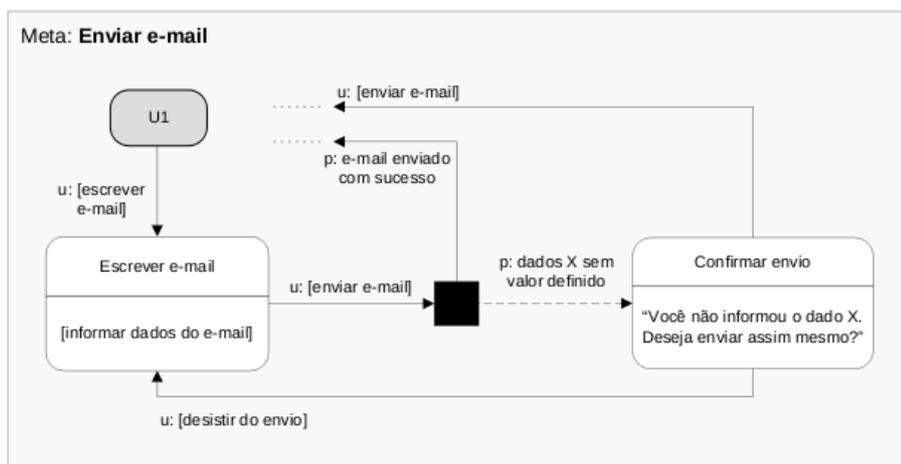


Figura 3.2: Uma meta representada no diagrama de interação, tal como proposto em versões iniciais da MoLIC (fonte: [Silva, 2005])

No entanto, a implementação desenvolvida ao longo deste trabalho e apresentada no capítulo 6 traz uma solução diferente para a representação de metas no diagrama de interação: ao invés de circunscrever estaticamente os elementos como uma grande caixa, tal como mostrado na Figura 3.2, é mostrado um *overlay* sobre os objetos que pode ser ativado ou desativado. Isso visa proporcionar uma comparação mais clara entre metas diferentes, num mesmo diagrama (ver seção 6.4).

### 3.3

#### Esquema Conceitual de Signos

Dentro da perspectiva da Engenharia Semiótica, o designer precisa codificar no seu software os signos que julgar suficientes para passar a sua mensagem para o usuário de forma eficiente. Em outras palavras, o sistema de significação escolhido pelo designer deve fazer sentido para o usuário, na conversa.

Sendo que a conversa entre as duas partes se dá em termos de signos, a MoLIC provê um meio de organização e definição desses signos através do esquema conceitual de signos. Ele é feito em paralelo com a construção do diagrama de interação e serve para que o designer especifique, inicialmente, a **ontologia** dos signos. Ou seja, com parte dos signos e seus relacionamentos, sem necessariamente definir todos os atributos dos signos de antemão – à medida que o designer especifica os signos que compõem os diálogos, a ontologia de signos evoluiria e novos signos iriam sendo definidos [Silva, 2005].

Atualmente, o designer especifica os signos em tabelas: a identificação, o conteúdo, e mecanismos de prevenção e recuperação de rupturas e possíveis expressões de todos os signos conceituais envolvidos em cada diálogo. O trabalho de Silva detalha de forma mais abrangente a utilização desse esquema em [Silva, 2005]. Este trabalho, no entanto, não apoia a definição ou manipulação do esquema conceitual de signos.

### 3.4

#### Mapeamento em elementos de interface

Silva e Barbosa em [Silva e Barbosa, 2007] desenvolveram um guia prático para auxiliar designers na utilização da linguagem MoLIC. Neste trabalho apresentam também o mapeamento dos elementos da notação para elementos de interface, baseado em padrões emergentes durante a utilização da linguagem, como relatado em [Silva et al., 2005]. Assim os mapeamentos podem ser relacionados como:

- a. **Acesso Ubíquo – menu ou barra de navegação.** Como significam que “a partir de qualquer ponto da aplicação o usuário pode dizer X para iniciar uma conversa sobre um determinado tópico”, acessos ubíquos podem ser mapeados para um item em um menu (no caso de aplicações de *Desktop*) ou em uma barra de navegação (no caso de *websites*);

- b. **Fala do usuário – botão ou link.** Quando uma fala do usuário leva a um processamento de sistema, geralmente é mapeado em um botão. No caso em que a fala leva diretamente a uma outra cena (como na navegação de um website), o mapeamento costuma ser feito para um *link*;
- c. **Cena – unidade de apresentação (tela).** Durante o projeto de interação, podemos considerar um mapeamento 1-para-1 entre uma cena e uma unidade de apresentação como uma janela em uma aplicação *Desktop* ou uma página em um *website*. No entanto este mapeamento pode variar, pois várias cenas podem ser agrupadas numa mesma unidade de apresentação (p.ex. dispositivos com *displays* grandes) ou uma cena pode possuir/estar fragmentada em várias unidades de apresentação (p.ex. em dispositivos móveis com *displays* de espaço limitado);
- d. **Fala do designer – unidade de apresentação separada ou um *feedback* na própria unidade.** Em aplicações *Desktop*, geralmente a fala do designer é representada como uma mensagem em uma caixa de alerta ou em uma barra de status. Quando a mensagem precisa de uma intervenção posterior do usuário, o mapeamento pode ser em uma janela de diálogo separada, com as respectivas falas do usuário partindo dela. Porém, quando se trata apenas de uma mensagem a ser lida pelo usuário, a fala do designer leva a uma cena que a possui, implicitamente;
- e. **Diálogo – grupos de signos relacionados.** Os diálogos representam subtópicos de uma cena, portanto, podem ser estruturados dentro de uma unidade de apresentação em forma de painéis ou qualquer outra estratégia de agrupamento de signos;

Apesar da relação entre os elementos da linguagem e elementos de interface, a ferramenta apresentada neste trabalho não apoia o mapeamento da MoLIC em elementos de interface – isto representa um passo adiante no processo apoiado pela ferramenta e pela notação, ficando este a cargo do designer, ao progredir para a implementação da solução.